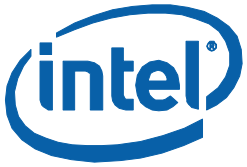

Intel® MPI Library for Windows* OS

User's Guide

Copyright © 2003–2014 Intel Corporation

All Rights Reserved

Document Number: 316404-011

Contents

1. Introduction 3
    1.1. Introducing Intel® MPI Library 3
    1.2. Intended Audience 3
    1.3. Notational Conventions 3
    1.4. Related Information 4
2. Using Intel® MPI Library 5
    2.1. Usage Model 5
    2.2. Before You Begin 5
    2.3. Quick Start 6
    2.4. Compiling and Linking 7
    2.5. Setting up Hydra Services 7
    2.6. Selecting a Network Fabric 8
        2.6.1. I_MPI_FABRICS 8
    2.7. Running an MPI Program 9
    2.8. Controlling MPI Process Placement 9
3. Troubleshooting 11
    3.1. Testing Installation 11
    3.2. Compiling and Running a Test Program 11

Disclaimer and Legal Notices

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

A "Mission Critical Application" is any application in which failure of the Intel Product could result, directly or indirectly, in personal injury or death. SHOULD YOU PURCHASE OR USE INTEL'S PRODUCTS FOR ANY SUCH MISSION CRITICAL APPLICATION, YOU SHALL INDEMNIFY AND HOLD INTEL AND ITS SUBSIDIARIES, SUBCONTRACTORS AND AFFILIATES, AND THE DIRECTORS, OFFICERS, AND EMPLOYEES OF EACH, HARMLESS AGAINST ALL CLAIMS COSTS, DAMAGES, AND EXPENSES AND REASONABLE ATTORNEYS' FEES ARISING OUT OF, DIRECTLY OR INDIRECTLY, ANY CLAIM OF PRODUCT LIABILITY, PERSONAL INJURY, OR DEATH ARISING IN ANY WAY OUT OF SUCH MISSION CRITICAL APPLICATION, WHETHER OR NOT INTEL OR ITS SUBCONTRACTOR WAS NEGLIGENT IN THE DESIGN, MANUFACTURE, OR WARNING OF THE INTEL PRODUCT OR ANY OF ITS PARTS.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined". Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725, or go to: http://www.intel.com/design/literature.htm

MPEG-1, MPEG-2, MPEG-4, H.261, H.263, H.264, MP3, DV, VC-1, MJPEG, AC3, AAC, G.711, G.722, G.722.1, G.722.2, AMRWB, Extended AMRWB (AMRWB+), G.167, G.168, G.169, G.723.1, G.726, G.728, G.729, G.729.1, GSM AMR, GSM FR are international standards promoted by ISO, IEC, ITU, ETSI, 3GPP and other organizations. Implementations of these standards, or the standard enabled platforms may require licenses from various entities, including Intel Corporation.

BlueMoon, BunnyPeople, Celeron, Celeron Inside, Centrino, Centrino Inside, Cilk, Core Inside, E-GOLD, Flexpipe, i960, Intel, the Intel logo, Intel AppUp, Intel Atom, Intel Atom Inside, Intel Core, Intel Inside, Intel Insider, the Intel Inside logo, Intel NetBurst, Intel NetMerge, Intel NetStructure, Intel SingleDriver, Intel SpeedStep, Intel Sponsors of Tomorrow., the Intel Sponsors of Tomorrow. logo, Intel StrataFlash, Intel vPro, Intel XScale, InTru, the InTru logo, the InTru Inside logo, InTru soundmark, Itanium, Itanium Inside, MCS, MMX, Moblin, Pentium, Pentium Inside, Puma, skoool, the skoool logo, SMARTi, Sound Mark, Stay With It, The Creators Project, The Journey Inside, Thunderbolt, Ultrabook, vPro Inside, VTune, Xeon, Xeon Inside, X-GOLD, XMM, X-PMU and XPOSYS are trademarks of Intel Corporation in the U.S. and/or other countries.

* Other names and brands may be claimed as the property of others.

Microsoft, Windows, and the Windows logo are trademarks, or registered trademarks of Microsoft Corporation in the United States and/or other countries.

Java is a registered trademark of Oracle and/or its affiliates.

Copyright (C) 2003–2014, Intel Corporation. All rights reserved.

Optimization Notice

Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

Notice revision #20110804

1. Introduction

This User's Guide explains how to use the Intel® MPI Library to compile and run a simple MPI program. This guide also includes basic usage examples and troubleshooting tips.

To quickly start using the Intel® MPI Library, print this short guide and walk through the example provided.

The *Intel® MPI Library for Windows* OS User's Guide* contains information on the following subjects:

- First steps using the Intel® MPI Library
- First-aid troubleshooting actions

This *User's Guide* contains the following sections:

Document Organization

Section	Description
Section 1 Introduction	Section 1 introduces this document
Section 2 Using the Intel® MPI Library	Section 2 describes how to use the Intel® MPI Library
Section 3 Troubleshooting	Section 3 outlines first-aid troubleshooting actions

1.1. Introducing Intel® MPI Library

The Intel® MPI Library is a multi-fabric message passing library that implements the Message Passing Interface, version 3.0 (MPI-3.0) specification.

1.2. Intended Audience

This *User's Guide* is intended for first time users of Intel® MPI Library.

1.3. Notational Conventions

The following conventions are used in this document.

Table 1.3-1 Conventions and Symbols used in this Document

This type style	Document or product names
This type style	Hyperlinks
`This type style`	Commands, arguments, options, file names
`THIS_TYPE_STYLE`	Environment variables
`<this type style>`	Placeholders for actual values
`[ items ]`	Optional items

`{ item | item }`	Selectable items separated by vertical bar(s)
(SDK only)	For Software Development Kit (SDK) users only

1.4. Related Information

To get more information about the Intel® MPI Library, explore the following resources:

The *Intel® MPI Library Release Notes* include key product details. See the *Intel® MPI Library Release Notes* for updated information on requirements, technical support, and known limitations. Use the Windows* OS program menu to select Intel(R) Software Development Tools > Intel(R) MPI Library 5.0 > Intel(R) MPI Library for Windows* OS Release Notes.

Visit the *Intel® MPI Library for Windows* OS Knowledge Base* for additional troubleshooting tips and tricks, compatibility notes, known issues, and technical notes.

For more information see Websites:

Product Web Site

Intel® MPI Library Support

Intel® Cluster Tools Products

Intel® Software Development Products

2. Using Intel® MPI Library

This section describes the basic Intel® MPI Library usage model and demonstrates typical usages of the Intel® MPI Library.

2.1. Usage Model

Using the Intel® MPI Library involves the following steps:

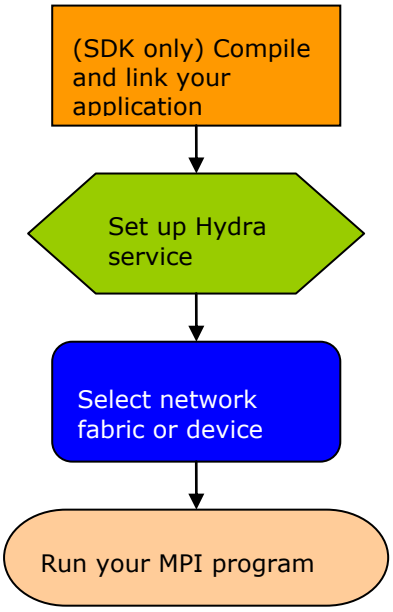

Figure 1: Flowchart representing the usage model for working with the Intel® MPI Library.

2.2. Before You Begin

1. Before using the Intel® MPI Library, ensure that the library, scripts, and utility applications are installed. See the *Intel® MPI Library for Windows* OS Installation Guide* for installation instructions.

2. For getting proper environment settings, use the following commands from the Start menu:

   Start >
   Programs >
   Intel(R) Software Development Products>
   Intel(R) MPI Library 5.0>
   Build Environment for the Intel® 64>
   Build Environment

   Alternatively, you can open a new console (cmd) window and run the following BAT file from the command line:

   - `<install_dir>\intel64\bin\mpivars.bat`

3. Ensure that you have administrator privileges on all nodes of the cluster to start the `Hydra` service on all nodes of the cluster.

2.3. Quick Start

To start using the Intel® MPI Library:

1. Set the environment settings: use the `call` batch command to run the `mpivars.bat` batch scripts in the `<installdir>\intel64\bin` directory for the Intel® 64 architecture.

2. Ensure the `Hydra` services are installed and started on compute nodes:
   a. To install the `Hydra` services from the command line, use the `hydra_service -install` commmand.
   b. To start the `Hydra` service, go to **Computer Management** -> **Services and Applications** -> **Services**, or from the command line manually using the `hydra_service -start` command.

To compile your MPI program:

1. **(SDK only)** Ensure that you have a compiler in your `PATH`.

2. **(SDK only)** Compile the test program using the appropriate compiler driver. For example:

```
> mpicc.bat -o test <installdir>\test\test.c
```

3. Register your credentials using the `wmpiregister` GUI utility, or the `mpiexec -register` command line option.

To run your MPI program:

1. Execute the test:

   a. Using the GUI utility `wmpiexec`. Set the application name and the number of processes. The test runs all processes on the current host. To start a test on a remote host, or on more than one host, press the **Advanced Options** button and fill the appropriate fields. Use the **Show Command** button to check the command line. Press the **Execute** button to start the program.

   b. Use the CLI `mpiexec` command to execute the test.

```
> mpiexec.exe -n <# of processes> test.exe
```

   or

```
> mpiexec.exe -hosts <# of hosts> <host1_name>
<host1 # of processes> <host2_name> \
<host2 # of processes> ... test.exe
```

NOTE: You can also run the MPI program in the runtime environment.

See the rest of this document and the *Intel® MPI Library Reference Manual for Windows*OS* for more details.

2.4. Compiling and Linking

(SDK only)

To compile and link an MPI program with the Intel® MPI Library do the following steps:

1. Create a Winxx Console project for Microsoft* Visual Studio*.
2. Choose the x64 solution platform.
3. Add `<installdir>\intel64\include` to the include path.
4. Add `<installdir>\intel64\lib` to the library path.
5. Add `impi.lib` (Release) or `impid.lib` (Debug) to your target link command for C applications.
6. Add `impi.lib` and `impicxx.lib` (Release) or `impid.lib` and `impicxxd.lib` (Debug) to your target link command for C++ applications.
7. Link application with `impimt.lib` (Release) or `impidmt.lib` (Debug) for multithreading.
8. Build a program.
9. Place your application and all the dynamic libraries in a shared location or copy them to all the nodes.
10. Run the application using the `mpiexec.exe` command.

2.5. Setting up Hydra Services

Intel® MPI Library uses a scalable process management system (Hydra) job startup mechanism. To run programs compiled with Microsoft* Visual Studio* (or related), set up an HYDRA service.

NOTE

You should have the administrator privileges to start the `Hydra` service and all users can launch processes with `mpiexec`.

To set up HYDRA services:

1. During the Intel® MPI Library installation the `Hydra` service is started. During installation you can cancel the `Hydra` service startup.
2. You can start, restart, stop or remove the `Hydra` service manually when the Intel® MPI Library is installed. Find `hydra_service.exe` in the `<installdir>\intel64\bin`.
3. Use the following command on each node of the cluster: `hydra_service.exe -remove` to remove the previous `Hydra` service.
4. Use the following command on each node of the cluster: `hydra_service.exe -install` to install the `Hydra` service manually.

2.6. Selecting a Network Fabric

The Intel® MPI Library dynamically selects different fabrics for communication between MPI processes. To select a specific fabric combination, set the `I_MPI_FABRICS` environment variable.

2.6.1. I_MPI_FABRICS

Select a particular network fabric to be used for communication.

Syntax

```
I_MPI_FABRICS=<fabric>|<intra-node fabric>:<inter-nodes fabric>
```

Where

- `<fabric> := {shm, dapl, tcp}`
- `<intra-node fabric> := {shm, dapl, tcp}`
- `<inter-nodes fabric> := {dapl,tcp}`

Arguments

Argument	Definition
<fabric>	Define a network fabric
`shm`	Shared-memory
`dapl`	DAPL-capable network fabrics, such as InfiniBand*, iWarp*, Dolphin*, and XPMEM* (through DAPL*)
`tcp`	TCP/IP-capable network fabrics, such as Ethernet and InfiniBand* (through IPoIB*)

For example, to select the OFED* InfiniBand* device, use the following command:

```
> mpiexec.hydra -n <# of processes>  \
-env I_MPI_FABRICS shm:dapl <executable>
```

For these devices, if *<provider>* is not specified, the first DAPL* provider in the `dat.conf` file is used. The `shm` fabric is available for both Intel® and non-Intel microprocessors, but it may perform additional optimizations for Intel microprocessors than it performs for non-Intel microprocessors.

NOTE

Ensure the selected fabric is available. For example, use `shm` only if all the processes can communicate with each other through shared memory. Use `dapl` only if all the processes can communicate with each other through a single DAPL provider. Ensure that the `dat.dll` library is in your `%PATH%`. Otherwise, use the `-genv` option for `mpiexec.exe` for setting the `I_MPI_DAT_LIBRARY` environment variable with the fully-qualified path to the `dat.dll` library.

2.7. Running an MPI Program

To launch programs linked with the Intel® MPI Library use the `mpiexec` command:

```
> mpiexec.exe -n <# of processes> myprog.exe
```

The `wmpiexec` utility is a GUI wrapper for `mpiexec.exe`. See the *Intel® MPI Library Reference Manual* for more details.

To set the number of processes on the local node, use the only required `mpiexec -n` option.

To set names of hosts and number of processes, use the `-hosts` option :

```
> mpiexec.exe -hosts 2 host1 2 host2 2 myprog.exe
```

If you are using a network fabric as opposed to the default fabric, use the `-genv` option to set the `I_MPI_FABRICS` variable.

For example, to run an MPI program using the `shm` fabric, type in the following command:

```
> mpiexec.exe -genv I_MPI_FABRICS shm -n <# of processes> \

myprog.exe
```

To run the program, you may use the `-configfile` option :

```
> mpiexec.exe -configfile config_file
```

The configuration file contains:

```
-host host1 -n 1 -genv I_MPI_FABRICS shm:dapl myprog.exe

-host host2 -n 1 -genv I_MPI_FABRICS shm:dapl myprog.exe
```

For the `rdma` capable fabric, use the following command:

```
> mpiexec.exe -hosts 2 host1 1 host2 1 -genv I_MPI_FABRICS dapl myprog.exe
```

You can select any supported device. For more information, see Section *Selecting a Network Fabric.*

If you successfully run your application using the Intel® MPI Library, you can move your application from one cluster to another and use different fabrics between the nodes without re-linking. If you encounter problems, see *Troubleshooting* for possible solutions.

2.8. Controlling MPI Process Placement

The `mpiexec` command controls how the ranks of the processes are allocated to the nodes of the cluster. By default, the `mpiexec` command uses group round-robin assignment, putting consecutive MPI process on all processor ranks of a node. This placement algorithm may not be the best choice for your application, particularly for clusters with symmetric multi-processor (SMP) nodes.

Suppose that the geometry is `<#ranks> = 4` and `<#nodes> = 2`, where adjacent pairs of ranks are assigned to each node (for example, for two-way SMP nodes). To see the cluster nodes, enter the command:

```
type machines.Windows
```

The results should look as follows:

```
clusternode1
clusternode2
```

To equally distribute four processes of the application on two-way SMP clusters, enter the following command:

```
mpiexec -n 2 -host clusternode1 .\testf : -n 2  -host clusternode2 .\testf
```

The output for the `testf` executable file may look as follows:

```
Hello world: rank 0 of 4 running on clusternode1

Hello world: rank 1 of 4 running on clusternode1

Hello world: rank 2 of 4 running on clusternode2

Hello world: rank 3 of 4 running on clusternode2
```

In general, if a cluster has `i` nodes and each node is a `j`-way SMP system, the `mpiexec` command-line syntax for distributing the `i*j` processes amongst the `i*j` processors within the cluster is:

```
mpiexec -n j -host <nodename-1> .\mpi_example :  \

-n  j -host <nodename-2> .\mpi_example : \

-n  j -host <nodename-3> .\mpi_example : \

...

-n  j -host <nodename-i> .\mpi_example
```

NOTE: You have to fill in appropriate host names from `<nodename-1>` through `<nodename-i>` with respect to your cluster system.

See Also

Local Options topic of the *Intel® MPI Library Reference Manual for Windows OS*.

You can get more information about controlling MPI process placement at Controlling Process Placement with the Intel® MPI Library.

3. Troubleshooting

This section explains how to test the Intel® MPI Library installation and how to run a test program.

3.1. Testing Installation

To ensure that the Intel® MPI Library is installed and functioning, complete a general testing, compile and run a test program.

To test the installation:

1. Verify through the Computer Management that the `Hydra` service is started. It calls the Intel MPI Process Manager.

2. Verify that `<installdir>\intel64\bin` for the Intel® 64 architecture in the 64-bit mode is in your path:

```
> echo %PATH%
```

You should see the correct path for each node you test.

3. **(SDK only)** If you use Intel® compilers, verify that the appropriate directories are included in the `PATH` and `LIB` environment variables:

```
> mpiexec.exe -hosts 2 host1 1 host2 1 a.bat
```

where `a.bat` contains

```
echo %PATH%
```

You should see the correct directories for these path variables for each node you test. If not, call the appropriate `compilervars.bat` scripts. For example, with Intel® C++ Compiler 14.0 for Windows*OS for the Intel® 64 architecture in the 64-bit mode, use the Windows OS program menu to select:
Intel(R) Parallel Studio XE 2013 > Command Prompt > Intel 64 Visual Studio 2012 mode or from the command line

```
%ProgramFiles%\Intel\Composer XE\bin\iclvars.bat
```

3.2. Compiling and Running a Test Program

The install directory `<installdir>\test` contains test programs which you can use for testing. To compile one of them or your test program, do the following:

1. **(SDK only)** Compile a test program as described in Section 2.4 Compiling and Linking.

2. If you are using InfiniBand* or other RDMA-capable network hardware and software, verify that everything is functioning.

3. Run the test program with all available configurations on your cluster.

4. Test the TCP/IP-capable network fabric using:
```
$ mpiexec -n 2 -env I_MPI_DEBUG 2 -env I_MPI_FABRICS tcp ./myprog
```

You should see one line of output for each rank, as well as debug output indicating the TCP/IP-capable network fabric is being used.

5. Test the shared-memory and DAPL-capable network fabrics using:

   ```
   $ mpiexec -n 2 -env I_MPI_DEBUG 2 -env I_MPI_FABRICS shm:dapl ./myprog
   ```
   You should see one line of output for each rank, as well as debug output indicating the shared-memory and DAPL-capable network fabrics are being used.

6. Test any other fabric using:
   `mpiexec -n 2 -env I_MPI_DEBUG 2 -env I_MPI_FABRICS <fabric> ./myprog`
   where `<fabric>` can be a supported fabric. For more information, see *Selecting a Network Fabric*.

For each of the `mpiexec` commands used, you should see one line of output for each rank, as well as debug output indicating which device was used. The device(s) should agree with the `I_MPI_FABRICS` setting.